

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-288285

(43) 公開日 平成11年(1999)10月19日

(51) Int.Cl.<sup>8</sup>G 1 0 H 1/02  
7/02

識別記号

F I

G 1 0 H 1/02  
7/00

5 2 1 Z

審査請求 未請求 請求項の数21 O L (全 16 頁)

(21) 出願番号 特願平11-23285  
 (62) 分割の表示 特願平7-254366の分割  
 (22) 出願日 平成7年(1995)10月2日

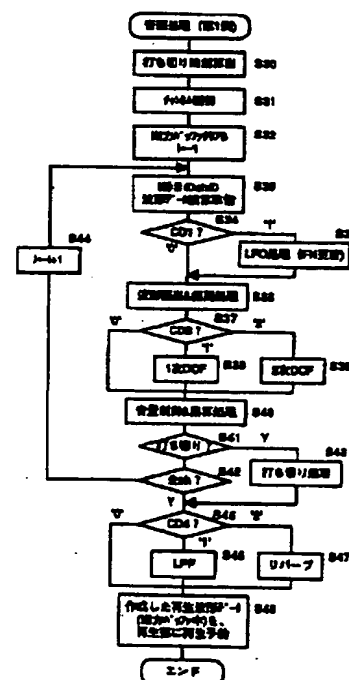
(71) 出願人 000004075  
 ヤマハ株式会社  
 静岡県浜松市中沢町10番1号  
 (72) 発明者 田邑 元一  
 静岡県浜松市中沢町10番1号 ヤマハ株式  
 会社内  
 (74) 代理人 弁理士 飯塚 義仁

(54) 【発明の名称】 楽音発生方法及び装置

(57) 【要約】

【課題】 処理している楽音の内容や並行して動作しているアプリケーションの内容などに基づいて楽音波形生成演算の内容を変更する。

【解決手段】 CPUが他のアプリケーションとマルチタスクでソフト音源を実行して楽音生成処理を行う。ソフト音源の実行にあたって、演奏情報を受信するとともに選択情報を受信し、演奏情報に基づく演算処理によって楽音を生成するとともに、選択情報に応じて楽音生成における所定の特性制御処理を追加または省略する。これにより、同時生成可能な最大発音数を変化させ、また同一の演奏情報に対しても異なる演算処理が実行され得るようになる。また、演算量の上限を示す制限情報を受信し、同時生成する楽音数を、該上限に応じて定まる最大同時発音数以下になるように各楽音の優先度に従って制限する。更に、サンプリング周波数あるいは波形補間の種類を示す演算精度情報を受信し、演算精度情報が示すサンプリング周波数または波形補間の種類で楽音生成演算処理を行なう。



## 【特許請求の範囲】

【請求項 1】 プログラムに従って動作する CPU が、演算情報を受信するとともに選択情報を受信し、前記演算情報に基づく演算処理によって楽音を生成するとともに、前記選択情報に応じて楽音生成における所定の特性制御処理を追加または省略することにより、前記 CPU で同時に生成可能な最大発音数を変化させ、かつ、同一の演算情報に対しても異なる演算処理が実行され得ることを特徴とする楽音発生方法。

【請求項 2】 前記選択情報は、(1) 低周波発振器のオン・オフ、(2) 和音のオン・オフおよび／または和音の種類、(3) フィルタのオン・オフおよび／またはフィルタの種類、(4) エフェクトのオン・オフおよび／またはエフェクトの種類、(5) サンプリング周波数のうちの少なくとも 1 つについての選択情報である請求項 1 に記載の楽音生成方法。

【請求項 3】 プログラムに従って動作する CPU が、演算情報を受信するとともに当該 CPU の演算量の上限を示す制限情報を受信し、前記演算情報に基づく演算処理によって楽音を生成するとともに、前記 CPU で同時に生成する楽音数を、前記制限情報が示す演算量の上限に応じて定まる最大同時発音数以下になるように各楽音の優先度に従って制限することを特徴とする楽音発生方法。

【請求項 4】 前記制限情報を表示することを特徴とする請求項 3 に記載の楽音発生方法。

【請求項 5】 前記 CPU の演算量が前記制限情報が示す上限に達した際に、前記 CPU で同時に生成する楽音数を制限することを特徴とする請求項 3 に記載の楽音発生方法。

【請求項 6】 前記演算情報を新たに受信した後に、前記 CPU で同時に生成する楽音数を制限する処理を行なうことを特徴とする請求項 3 に記載の楽音発生方法。

【請求項 7】 プログラムに従って動作する CPU が、演算情報を受信するとともに任意に設定されたサンプリング周波数を示す演算精度情報を受信し、前記演算情報に基づき、前記演算精度情報が示すサンプリング周波数の演算処理によって楽音を生成することを特徴とする楽音発生方法。

【請求項 8】 プログラムに従って動作する CPU が、演算情報を受信するとともに波形和音の種類を示す演算精度情報を受信し、前記演算情報に基づき、前記演算精度情報が示す種類の波形和音を含む演算処理によって楽音を生成することを特徴とする楽音発生方法。

【請求項 9】 前記演算精度情報を表示することを特徴とする請求項 7 または 8 に記載の楽音発生方法。

【請求項 10】 前記 CPU における前記演算処理の演算量およびその演算処理の結果生成された楽音数を表示することを特徴とする請求項 1 乃至 9 のいずれかに記載の楽音発生方法。

【請求項 11】 プログラムに従って動作する CPU であって、演算情報を受信可能にするとともに選択情報を受信可能とし、前記演算情報に基づいて、楽音生成演算を実行するとともに、前記選択情報に応じて楽音生成における所定の特性制御処理を追加または省略することにより、前記 CPU で同時に生成可能な最大発音数を変化し、かつ、同一の演算情報に対しても異なる演算処理が実行され得るものを備えたことを特徴とする楽音発生装置。

【請求項 12】 前記選択情報は、(1) 低周波発振器のオン・オフ、(2) 和音のオン・オフおよび／または和音の種類、(3) フィルタのオン・オフおよび／またはフィルタの種類、(4) エフェクトのオン・オフおよび／またはエフェクトの種類、(5) サンプリング周波数のうちの少なくとも 1 つについての選択情報である請求項 11 に記載の楽音発生装置。

【請求項 13】 プログラムに従って動作する CPU であって、演算情報を受信可能にするとともに当該 CPU の演算量の上限を示す制限情報を受信可能とし、前記演算情報に基づく演算処理によって楽音を生成するとともに、前記 CPU で同時に生成する楽音数を、前記制限情報が示す演算量の上限に応じて定まる最大同時発音数以下になるように各楽音の優先度に従って制限するものを備えたことを特徴とする楽音発生装置。

【請求項 14】 前記制限情報を表示する表示手段を備えたことを特徴とする請求項 13 に記載の楽音発生装置。

【請求項 15】 前記 CPU は、当該 CPU の演算量が前記制限情報が示す上限に達した際に、前記 CPU で同時に生成する楽音数を制限することを特徴とする請求項 13 に記載の楽音発生装置。

【請求項 16】 前記 CPU は、当該 CPU が前記演算情報を新たに受信した後に、当該 CPU で同時に生成する楽音数を制限する処理を行なうことを特徴とする請求項 13 に記載の楽音発生装置。

【請求項 17】 プログラムに従って動作する CPU であって、演算情報を受信可能にするとともに任意に設定されたサンプリング周波数を示す演算精度情報を受信可能にし、前記演算情報に基づき、前記演算精度情報が示すサンプリング周波数の演算処理によって楽音を生成するものを備えたことを特徴とする楽音発生装置。

【請求項 18】 プログラムに従って動作する CPU であって、演算情報を受信可能にするとともに波形和音の種類を示す演算精度情報を受信可能にし、前記演算情報に基づき、前記演算精度情報が示す種類の波形和音を含む演算処理によって楽音を生成するものを備えたことを特徴とする楽音発生装置。

【請求項 19】 前記演算精度情報を表示する表示手段を備えたことを特徴とする請求項 17 または 18 に記載の楽音発生装置。

【請求項 20】 前記 CPU における前記演算処理の演算量およびその演算処理の結果生成された楽音数を表示することを特徴とする請求項 11 乃至 19 のいずれかに記載の楽音発生装置。

【請求項 21】 前記 CPU は、所定周期毎に楽音の複数サンプルを一括して生成することを特徴とする請求項 1 乃至 20 のいずれかに記載の楽音発生方法または装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、CPU（中央処理装置）や DSP（ディジタル信号処理装置）などのプログラマブルな演算装置上で楽音生成プログラムを実行することにより楽音波形を生成する楽音発生方法に関し、さらに、同楽音生成プログラムを実行することによって楽音波形を生成する楽音発生装置に関する。

【0002】

【従来の技術】従来の音源装置や演算によって楽音波形を生成する楽音生成プログラムでは、サンプリング周波数や最大発音数および各楽音の処理内容は、どのような楽音を生成する場合、また、他の処理がどのような状況であっても、予め設定されている条件で処理されていた。

【0003】

【発明が解決しようとする課題】しかし、このような方式では、（1）楽音生成演算が固定であるため、場合によっては必要以上の処理が行われていたり、必要な処理が含まれていなかったりという不都合があった。すなわち、楽音の種類によっては、楽音波形のピッチ変換が必要な場合と不要な場合があり、LFO（低周波発振器）による変調が必要な場合と不要な場合があり、ディジタルフィルタによる音色加工が必要な場合と不要な場合があり、エフェクトが必要な場合と不要な場合がある。しかし、従来の音源では、各回路が固定された処理を行っているため、新たな処理の追加、不要な処理の削除は、実現が難しく、実現のためには複雑な回路を付与する必要があった。

【0004】（2）ソフトウェアで音源を実現する場合、CPU の演算量は、発音中のチャンネル数や楽音生成演算の内容に応じてダイナミックに変動する。汎用コンピュータで他のアプリケーションプログラム（以下、「アプリケーション」という。）と並行してソフトウェア音源プログラム（以下、「ソフト音源」という。）を実行する場合、このソフト音源の処理量の変動（特に演算量の増加）により、他のアプリケーションの動作が不安定になることがあった。

【0005】（3）ソフト音源の処理に割り当てることができる演算量は、上記のように並行しているアプリケーションの数や種類のほか、それを実行する演算装置の演算能力により制限を受ける。割り当てられる演算量が

厳しく制限される場合でも、従来のソフトウェア音源プログラムでは、生成演算が固定的に決定されていたため、ユーザが、生成演算クオリティを落としても発音数を増やしたい場合や、発音数は少なくともよいので高いクオリティで生成演算したい場合等の選択を行うことができなかった。

【0006】この発明は、処理している楽音の内容や並行して動作しているアプリケーションの内容などに基づいて楽音波形生成演算の内容を変更することのできる楽音生成方法および楽音生成装置を提供することを目的とする。

【0007】

【課題を解決するための手段】この発明に係る楽音発生方法は、プログラムに従って動作する CPU が、演奏情報を受信するとともに選択情報を受信し、前記演奏情報に基づく演算処理によって楽音を生成するとともに、前記選択情報に応じて楽音生成における所定の特性制御処理を追加または省略することにより、前記 CPU で同時に生成可能な最大発音数を変化させ、かつ、同一の演奏情報に対しても異なる演算処理が実行され得ることを特徴とするものである。

【0008】この発明によれば、プログラムに従って動作する CPU により、演奏情報の受信と選択情報の受信が行なわれ、受信した演奏情報に基づく楽音生成処理が行なわれるとともに、選択情報に応じて該楽音生成における所定の特性制御処理を追加または省略することにより、当該 CPU で同時に生成可能な最大発音数を変化させることができ、かつ、同一の演奏情報に対しても異なる演算処理が実行され得るようになる。すなわち、選択情報に応じて所定の特性制御処理を追加したときと省略したときとでは、同一の演奏情報に基づく楽音生成を行なうのであっても、異なる演算処理が適宜実行されることとなる。

【0009】従って、生成しようとする楽音の目的や演奏形態に適合させて、楽音制御の内容を変更したり、特定の楽音制御をやめることで処理量を軽減し、その分発音数を増加させたり、他の目的に CPU のパワーをまわしたりすることができる。これによって、例えば、マイクロコンピュータを用いたソフト音源を実施する場合、コンピュータ装置の CPU パワーを有効に活用することができる。また、他のアプリケーションプログラムと並行してソフトウェア音源プログラムを実行する場合、ソフト音源側の演算量を調整することができるので、並行して動作する他のアプリケーションプログラムに対して悪影響を与えないようにすることもできる。このように、選択情報に応じて必要な特性制御処理を追加しまた不要な特性制御処理を削除することにより、その都度の楽音生成に適した楽音波形生成制御が可能になり、また、不要な処理を削減することによって演算量が軽減されるため、発音数を増加したり、楽音生成以外の処理を

行うこともできる。従って、ソフト音源と並行して他のアプリケーションを安定して動作させる場合にも適している。

【0010】一例として、選択情報は、(1) LFO (低周波発振器) のオン・オフ、(2) 補間のオン・オフおよび/または補間の種類、(3) フィルタのオン・オフおよび/またはフィルタの種類、(4) エフェクトのオン・オフおよび/またはエフェクトの種類、(5) サンプリング周波数、のうちの少なくとも1つについての選択情報である。

【0011】この発明の別の観点に従う楽音発生方法は、プログラムに従って動作するCPUが、演奏情報を受信するとともに当該CPUの演算量の上限を示す制限情報を受信し、前記演奏情報に基づく演算処理によって楽音を生成するとともに、前記CPUで同時に生成する楽音数を、前記制限情報が示す演算量の上限に応じて定まる最大同時発音数以下になるように各楽音の優先度に従って制限することを特徴とするものである。この発明によれば、演算量を制限しつつも、その限られた演算量のなかで生成できた楽音については音切れ等起こさず安定した発音が行える。また、発音数の制限を行なうに際し、各楽音の優先度に従って制限するので、優先度の高い楽音が消音されないこととなり、非常に好ましい。従って、他のアプリケーションプログラムと並行してソフトウェア音源プログラムを実行する場合、ソフト音源がCPUパワーを占有する時間を制限情報で制限することにより、円滑なマルチタスク状態を確保するのに適している。

【0012】さらに、この発明の別の観点に従う楽音発生方法は、プログラムに従って動作するCPUが、演奏情報を受信するとともに任意に設定されたサンプリング周波数を示す演算精度情報を受信し、前記演奏情報に基づき、前記演算精度情報が示すサンプリング周波数の演算処理によって楽音を生成することを特徴とするものである。サンプリング周波数を示す演算精度情報とは、たとえば、等価サンプリング周波数(単位時間あたりに演算生成する楽音サンプル数)であり、この精度を粗くすることにより演算量を少なくすることができる。このようにサンプリング周波数を任意に設定することによって演算量を任意に可変設定することができ、これによって、臨機応変に、効率的なCPU資源の利用を図ることができる。

【0013】この発明のさらに別の観点に従う楽音発生方法は、プログラムに従って動作するCPUが、演奏情報を受信するとともに波形補間の種類を示す演算精度情報を受信し、前記演奏情報に基づき、前記演算精度情報が示す種類の波形補間を含む演算処理によって楽音を生成することを特徴とするものである。このように波形補間の精度を可変制御することによっても演算量を少なくする制御を行なうことができ、効率的なCPU資源の利

用を図ることができる。これらの発明は方法の発明として実施することができるのみならず、装置の発明として実施することができる。

【0014】

【発明の実施の形態】以下、図面を参照してこの発明の実施形態について説明する。図1はこの発明の一実施例に係るソフト音源プログラムが実行されるマイクロコンピュータ装置の構成を示す図である。CPU10にはバスを介してROM11、RAM12、ハードディスク装置13、CD-ROM装置14、MIDIインタフェース15、キーボード16、ディスプレイ17、DMA18およびタイマ21が接続されている。ROM11はこのマイコン装置の動作に必須の基本プログラムなどが記憶されている。RAM12は実行するプログラムやデータを読み込むとともにプログラム処理中に発生したデータを記憶するメモリである。ハードディスク装置13には、各種のアプリケーションなどが記憶されている。CD-ROM装置14には、各種のデータやプログラムを記憶したCD-ROMがセットされる。後述するソフト音源もCD-ROMから供給されるものである。ハードディスク装置13やCD-ROMに記憶されているプログラムは実行されるときRAM12に読み出される。MIDIインタフェース15は、外部接続されるMIDIキーボードなどの演奏装置との間で演奏データや制御信号の送受信を行う。キーボード16、ディスプレイ17としては、通常のパーソナルコンピュータに一般的なキーボードやモニタが接続される。DMA(Direct Memory Access Controller)18はCPU10を介さずにRAM12から波形データを読み出してDAC19に出力する回路である。DAC19は、この波形データをアナログの楽音信号に変換してサウンドシステム20に出力する。サウンドシステム20はこの楽音信号を増幅して外部出力する。なお、タイマ21は一定時間毎にCPU10に対して割り込みをかけるとともに、DMA18にサンプリングクロックを供給するタイマである。

【0015】図2は同マイクロコンピュータ装置で実行されるソフト音源の時間的な処理の流れを説明する図である。このソフト音源は48kHzのサンプリング周波数で楽音波形データを生成するが、その楽音波形データ生成処理を128サンプル(1フレーム)毎に行う。あるフレームのタイムスロットに演奏入力があると、次のフレームでその演奏入力に対応する楽音波形データの算出処理を行い、さらに次のフレームでこの楽音波形データを48kHzの周期毎に1サンプルずつ読み出して楽音信号を形成する。したがって、演奏入力があったから実際に楽音が発音されるまで(または楽音が消音されるまで)は約2フレームの時間ずれが生じることになるが、1フレームが128サンプル(約2.67ミリ秒)であるため、その時間ずれはわずかである。なお、本実

施態様では、RAM 12 上に用意された波形テーブルに記憶された波形サンプルに基づいて楽音を生成するいわゆるテーブルルックアップ方式の楽音生成を行うソフト音源を説明する。

【0016】図3は前記ソフト音源の動作時にRAM 12に設定される記憶エリアを説明する図である。同図(A)は入力バッファを示す図である。この入力バッファには、MIDIインタフェース15から演奏入力があったとき、その演奏入力の内容とその発生時刻を記憶するバッファである。このバッファの内容は、後述するMIDI処理で読み出され、対応する処理が実行される。

【0017】同図(B)はサンプルバッファWBを示す図、同図(C)は出力バッファOBを示す図である。両バッファとも128サンプル分の波形データ記憶エリア(SD1~SD128、OD1~OD128)を備えている。出力バッファOBは32チャンネルの発音チャンネルの楽音波形データを順次加算合成した波形データを記憶する。波形データの演算は、1つのチャンネル毎に1フレーム時間分の128サンプルを演算し、これを32チャンネル分(発音しているチャンネル分)繰り返すという手順で行われるが、1つのチャンネルの波形データを記憶するのがサンプルバッファWBであり、1つのチャンネルの波形データが演算されるごとにこの波形データをサンプルタイミング毎に累算してゆくのが出力バッファOBである。また、後述する他の実施態様では、選択可能な3つの等価サンプリング周波数(12kHz、24kHz、48kHz)に対応して3つの出力バッファOB0、OB1、OB2が存在する。

【0018】同図(D)は音色データレジスタである。この音色データレジスタには、各MIDIチャンネル(演奏パート)で生成される楽音波形を決定するデータが記憶されている。音色データとしては、各音色の各音域毎に素材とする波形テーブルを指定する各音域波形指定データ、EG制御データ、タッチ制御データなどがある。

【0019】同図(E)は音源レジスタである。この音源レジスタには、各発音チャンネル別に該発音チャンネルで生成される楽音波形を決定するデータが記憶される。データとしてはノートナンバ、いずれか1つの波形テーブルのアドレスを示す波形指定アドレス(アタックスタートアドレスAS、アタックエンドアドレスAE、ループスタートアドレスLS、ループエンドアドレスLE)、フィルタ制御データ、EG制御データ、ノートオンデータ、タイミングデータ、などが記憶される。

【0020】図4は前記ソフト音源が起動したとき、ディスプレイ17に表示される制御パネル画面を示す図である。この画面はディスプレイ17の画面の一部にいわゆるウィンドウ表示される。制御パネル画面にはMIDIモニタ31、LFOオン/オフ表示部33、補間(INT)設定表示部34、デジタルフィルタ(DCF)

設定表示部35、エフェクト(EFT)設定表示部36、サンプリング周波数(GSR)設定表示部37、最大発音数(MPF)設定表示部38、現在発音数表示部39、デューティ比表示部40、発音レベル表示部41の各種表示部が設けられるほか、カーソル(CURSOR)キー42、バリュウ(VALUE)スイッチ43、デューティ比(DUTY)スイッチ44およびリセット(RESET)スイッチ45の各種キースイッチが表示される。なお、背景色が濃くなっている部分32は、その位置にカーソルがあることを示している。

【0021】MIDIモニタ31はMIDIチャンネルにデータの入出力があったとき、そのチャンネルに対応するランプ部が点灯するモニタである。LFOは楽音にビブラートのようなうねり効果を生じさせるための低周波発信器である。この表示部33に、カーソルキー42を用いてカーソル32を移動させ、バリュウスイッチ43を操作することによってこのLFOのオン/オフを切り換えることができる。他の表示部34~38に関しても設定内容の切換方式は同様である。補間設定表示部34は、RAM 12の波形テーブルの波形データの周波数を生成する楽音の周波数に合わせてシフトする場合に生じるアドレスに小数部に対応したサンプルを得るため、それを何点補間するかを設定する表示部である。3次関数を用いる4点補間、1次関数を用いる2点補間、および、打楽器音など周波数シフトする必要のない楽音に対して設定される補間なしのいずれかを選択することができる。

【0022】デジタルフィルタ設定表示部35は、デジタルフィルタとして2次フィルタを用いるか1次フィルタを用いるかまたは音色フィルタを用いないかのいずれかを選択するための設定表示部である。エフェクト設定表示部36は、リバースまたはローパスフィルタ(LPF)のエフェクトをオン/オフするための設定表示部である。以上の設定表示部は、CPU 10の音源処理動作の処理内容を設定するための設定表示部である。

【0023】また、サンプリング周波数設定表示部37は、図11で後述する他の適用例に関する表示であり、このソフト音源で楽音波形データの演算を行う等価サンプリング周波数(通常は48kHz)を粗くしたい場合に設定変更される。48kHz以外に24kHz、12kHzのサンプリング周波数を設定することができる。さらに、最大発音数設定表示部38は、32音までの同時発音が可能なこのソフト音源を、CPU 10に対する負担を事前に軽減しておくため、最大発音数を32未満に抑える場合の最大発音数の表示である。その場合に、この最大発音数設定表示部38に表示される数値を変更(1~32)することにより、最大発音数をその値までにすることができる。

【0024】現在発音数表示部39は、現在ソフト音源が形成している楽音数である現在発音数を表示する。デ

ューティ比表示部40はCPU10の全能力のうち、このソフト音源が占有している能力の割合を表示する。能力は棒グラフ40aとして表示される。また、デューティ比スイッチ44を操作することにより、このソフト音源が占有するCPU10の能力のデューティ比の上限を示すデータDRを設定することができ、設定されたデューティ比はこのデューティ比表示部40に点線40bとして表示される。また、発音レベル表示部41は、現在の楽音信号の発音レベルを棒グラフ41aで示す表示部である。

【0025】フローチャートを参照して前記装置のマイクロコンピュータ装置におけるソフト音源の動作を説明する。図5(A)はメインルーチンを示すフローチャートである。プログラムが起動されると、まず、レジスタエリアの確保などの初期設定を実行したのち(S1)、図4に示す画面を準備する。そして何らかの起動要因

(トリガ)があるまでS3、S4で待機する。起動要因が発生した場合には、その起動要因をS5で判断して対応する処理動作を実行する。起動要因としては、入力バッファにMIDIデータが書き込まれた場合のMIDI処理(S6)、1フレームに対応する時間毎に実行される音源処理(S8)、その他スイッチイベントがあったときに実行されるその他処理(S10)、および、終了コマンドが入力された場合の終了処理(S12)がある。終了処理は設定データの退避やレジスタのクリアなどの処理であり、こののち、図4の画面を消去して(S13)、動作を終える。また、MIDI処理(S6)が行われたときは、MIDIモニタ31のそのMIDIデータを受信したチャンネルの表示器を点灯する(S7)。その他処理としては各種のパネル入力やコマンド入力に対応する処理があり、その一部は図10で説明する。この処理ののち、この処理に対応する表示変更処理を実行する(S11)。音源処理(S8)は、タイマ21が128サンプルクロックをカウントしたことによる割り込みまたはDMA18からのトリガ等によって図2における読出再生が次のフレームにち進行したのを検出して実行されるものであるが、図7、図8で詳細に説明する。また、デューティ比・発音数表示処理(P表示処理:S9)は図9で説明する。ここで、図7、図8は、ディスプレイ17の制御画面で、LFOのオン/オフ、補間設定、デジタルフィルタ設定、エフェクト設定ができるようにした場合例(第1実施態様)である。一方、図11は、ディスプレイ17の制御画面で、サンプリング周波数の設定ができるようにした例(第2実施態様)である。本明細書では、この他に、同制御画面において最大発音数を設定できるようにした場合の例(第3実施態様)についても簡単に説明する。

【0026】図5(B)のMIDI割込処理について説明する。この割込処理は最優先の割込処理動作である。MIDIインタフェース15からMIDIデータを受信

すると(S15)、該受信したMIDIデータとともに、その受信時刻データを入力バッファに書き込む(S16)。MIDI処理(S6:図6参照)は、入力バッファにMIDIデータが書き込まれていることを検出して起動する。MIDI処理では検出されたMIDIデータに対応した処理が行われる。

【0027】図6は、MIDI処理の1つであるノートオンイベント処理動作を示す図である。この処理動作は、入力バッファにノートオンイベントデータが書き込まれていたときに実行される動作である。まず、そのノートオンイベントデータのノートナンバ、ベロシティデータ、パート別音色および発生時刻をそれぞれNN、VEL、t、TMレジスタに記憶する(S20)。次に、32チャンネルの発音チャンネルのなかからこのノートオンに係る楽音を発音する発音チャンネルを割り当てiに記憶する(S21)。このノートオンに係る音色データTP(t)をNN、VELに応じて加工し(S22)、加工された音色データをノートオンを示すデータ、TMとともにiチャンネルの音源レジスタに書き込む(S23)。

【0028】図7はフレームに対応する時間周期で起動される音源処理を示すフローチャートである。まず、この音源処理動作の打ち切り時刻を算出する(S30)。打ち切り時刻は、このソフト音源処理にCPU10を占有できるデューティ比に基づいて算出され、楽音波形の形成(音源処理)に割くことができる時間を算出する処理である。すなわち、この音源処理動作では、32個の各発音チャンネルについて1フレーム(128サンプル)分の波形データを算出するが、演算途中でもCPU10の占有可能時間に達するとこの処理動作は強制的に終了する。途中で打ち切られた場合、一部のチャンネルについては波形データを算出できないが、この場合には、そのチャンネルの楽音はフォースダンプ(強制的に信号レベルを急速に減衰させて消音する動作)される。打ち切り時刻TL(タイムリミット)は、

$$TL = ST + FL \times DR - US - AS$$

ST: 現再生フレームのスタート時刻

FL: フレーム時間長(1フレーム全体の長さ)

DR: デューティ比

US: 打ち切り処理時間(打ち切り処理(フォースダンプ)に要する時間)

AS: 後処理時間(生成された複数チャンネル分の波形にリバーブ、LPF等の処理をしてDMAに再生予約し、音源処理を終えるまでの処理(後処理)に要する時間)

で計算される。

【0029】ここで、タイムリミットTLは、各フレームの音源処理中で実行される各チャンネルの楽音生成処理をいつまで続けて良いかを示すデータである。この式は、現フレームの開始時刻(ST)から終了時刻(ST

+FL)のうち、最初のデューティ比DRに応じた割合の時間を音源処理に使用可能とし、残りの時間でソフト音源のその他の処理や、その他のアプリケーションの処理を行えるようにした場合である。ここで、デューティ比DRは、デューティ比スイッチ44により設定されている。なお、この式では、打ち切り処理時間US、後処理時間ASを考慮に入れて、精密にタイムリミットを計算しているが、その演算は省略してもよい。

【0030】次にチャンネル制御を行う(S31)。チャンネル制御とは、上述したように、演算順が下位のチャンネルほど打ち切られる可能性が高いため、優先度の高いチャンネル(消音されては困るチャンネル)から先に演算するように32チャンネルの演算順序を設定する処理である。優先度の高いチャンネルとは、発音レベルの高いチャンネル、発音開始からの時間が短いチャンネルなどであり、優先度の低いチャンネルは、発音していないチャンネルを除けば、発音レベルの低いチャンネルが最も優先順位が低い。チャンネル制御ののち、出力バッファOBをクリアするとともに演算順位を示すポインタiに1をセットする(S32)。こののち、各発音チャンネルの波形データ演算処理(S33~S44)を実行する。

【0031】まず演算順がi番目のチャンネルの音源レジスタにアドレスを設定して、該チャンネルのデータを読み出し可能にするなどの波形データ演算準備処理を実行する(S33)。つぎに、LFO制御フラグCD1を判断する(S34)。CD1=1であれば、LFO処理(S35)を実行する。LFO処理とは、ノートナンバに対応したFナンバをLFO波形で周波数変調する処理(ビブラート処理)である。LFOの周波数は楽音周波数に比べて低く、また、1フレームが短い(128サンプル)ため、1フレームに一つの値でよい。もし、必要がない場合にはS34から直接S36に進む。S36では指定された波形テーブルからの波形読出および補間処理を実行する。Fナンバは、該ノートナンバの楽音を生成するため指定された波形テーブルをどの速さで読めばいいかを指定するデータであり、一般に整数部と小数部とからなる。

【0032】ここで、図8を参照してS36の波形読出・補間処理動作を説明する。この動作では、そのときiで指定されているチャンネルの波形データを1フレーム(128サンプル)分演算する。まず、サンプル数カウンタSに1をセットする(S50)。つぎに、補間方式レジスタCD2を参照して補間方式を判定する(S51)。ここで、CD2=0の場合は補間なし(ピッチ変換のないドラム音などを再生する場合)である。CD2=1の場合には2点補間(ラフな補間で済ませる場合)、CD3=2の場合には4点補間(折り返しの目立つデリケートな音の場合)である。

【0033】補間無しの場合には、まず、直前の演算の

アドレス(この場合、処理チャンネルの前のフレームの波形読み出しで最後に生成したアドレス)にFナンバを加算してアドレスの更新を行う(S52)。このようにして更新されたアドレスで指定された波形テーブルの波形サンプルを読み出しRDレジスタにセットする(S53)。RDの内容をサンプルバッファSD(S)にセットする(S54)。この動作をS=1からS=128になるまで繰り返し実行する(S55、S56)。128回の処理が完了すれば音源処理動作(図7)に戻る。この場合、Fナンバの小数部により生じたアドレスの小数部は無視されているので、このことにより折り返しノイズを生じる。また、Fナンバに小数部がなければこの問題は発生しない。

【0034】CD=1で2点補間をする場合には、まず、直前の演算のアドレスにFナンバを加算してアドレスの更新を行う(S57)。このとき、整数部と小数部からなるアドレスが生成されるため、指定された波形テーブルからこのアドレスを挟む2サンプル(整数部のアドレスで指定されるサンプルと整数部+1のアドレスで指定されるサンプル)の波形データを読み出す(S58)。これら2サンプルのデータを小数部の値で直線補間し、その値をIDレジスタにセットする(S59)。IDの内容をサンプルバッファSD(S)にセットする(S60)。この動作をS=1からS=128になるまで繰り返し実行する(S61、S62)。128回の処理が完了すれば音源処理動作(図7)に戻る。

【0035】CD=2で4点補間をする場合には、まず、直前の演算のアドレスにFナンバを加算してアドレスの更新を行う(S63)。このとき、整数部と小数部からなるアドレスが生成されるため、指定された波形テーブルからこのアドレスを挟む4サンプル(整数部-1、整数部、整数部+1、整数部+2のアドレスで指定されるサンプル)の波形データを読み出す(S64)。これら4サンプルのデータを通る3次曲線についてその曲線上の小数部に対応する点の値を求め、その値をIDレジスタにセットする(S65)。IDの内容をサンプルバッファSD(S)にセットする(S66)。この動作をS=1からS=128になるまで繰り返し実行する(S67、S68)。128回の処理が完了すれば音源処理動作(図7)に戻る。

【0036】図7において、波形読出・補間処理ののち、フィルタ制御レジスタCD3を参照する(S36)。CD3=0の場合にはデジタルフィルタ処理を行わないでそのまま音量制御・累算処理(S40)に進む。また、CD3=1の場合には、サンプルバッファWB(SD(1)~SD(128))の値にフィルタ制御データに応じた周波数特性を有する一次フィルタ処理を施す(S38)。また、CD3=2の場合には、サンプルバッファWBの値にフィルタ制御データに応じた周波数特性を有する二次フィルタ処理を施す(S39)。こののち音

量制御・累算処理動作に進む(S40)。

【0037】S40では、サンプルバッファWB(SD(1)～SD(128))の値に振幅EGおよびチャンネル音量パラメータに基づいて楽音の立ち上がりから立ち下がりにおける音量時間変化を付与する音量制御を施す。なお、振幅EGは一般的に緩やかな曲線であるため、128サンプルあたり1つのEG値でよい。このレベル制御されたサンプルバッファWBの値を出力バッファOB(OD(1)～OD(128))の対応するサンプルに足し込む。この加算動作が各i番目の演算順序を持つチャンネルについて順次繰返し実行されることにより、出力バッファにはそれまでに生成した全チャンネルの楽音波形データの累算値が記録されることになる。

【0038】こののち、処理打ち切り時刻TLになったか否かを判断する(S41)。打ち切り時刻になった場合には、打ち切り処理を実行する(S43)。打ち切り処理では、その時点までに波形データを演算できなかったチャンネルの楽音波形をフォースダンプするためのダンプ波形を生成し、出力バッファに足し込む。打ち切り時間になっていなければ発音中の全チャンネルについて処理を終了したかを判定する(S42)。すなわち、発音可能チャンネル数が32チャンネルであっても現在発音中のチャンネルがそれ以下であれば発音中のチャンネルに関する生成処理が終了すればS42の判断でS45以下に進む。まだ発音中の全チャンネルに関する処理が終了していなければ、演算順序を示すiに1を加算して(S44)、S33に戻る。

【0039】全チャンネルに関する処理が終了し、または、打ち切り処理が行われた場合にはS45に進行する。S45ではエフェクト制御レジスタCD4を参照する。CD4=0であればエフェクトをかけないためそのままS48に進む。CD=1であれば出力バッファOBの128サンプルの高域をカットするローパスフィルタLPFをかけるためS46でローパスフィルタ演算を施したのちS48に進む。また、CD4=2の場合には出力バッファOBの128サンプルに残響を付与するリバース処理をかけるためリバース演算を施したのち(S47)、S48に進む。S48では作成し出力バッファOBに記憶されている波形データを再生部に再生予約する。この再生予約は、RAM内の記憶アドレスをDMAに通知する処理である。

【0040】図9は、デューティ比・発音数表示処理(P表示処理)を示すフローチャートである。まず、S71でソフト音源全体または音源処理に費やされたCPU時間を算出する。算出された時間のCPU10の全体演算時間に対する比を算出し、これをCPU占有率表示部40に棒グラフで表示する(S72)。次に発音数iを現在発音数表示部39に表示する(S73)。

【0041】図10はその他処理のうち表示スイッチイベント処理ルーチンを示すフローチャートである。同図

(A)は、スイッチオンイベント処理ルーチンを示している。表示ウィンドウ30に表示されているスイッチ43がオンされた場合には、オンされたスイッチに応じて現在のカーソル32の位置に対応するデータCDxの値を設定する(S74)。このデータCDxは上記音源処理動作で利用される。

【0042】上記第1の実施態様では、カーソル32は、カーソルキー42の操作に応じて、図4に示される制御画面上のLFOオン/オフ表示部33、補間設定表示部34、デジタルフィルタ設定表示部35、エフェクト設定表示部36のいずれかの位置に移動可能である。バリュースイッチ43の操作に応じて、カーソル32の位置に対応したデータ、すなわち、LFOオン/オフ表示部33ではデータCD1、補間設定表示部34ではデータCD2、デジタルフィルタ設定表示部35ではデータCD3、エフェクト設定表示部36ではデータCD4の値が、それぞれ、設定変更される。同図(B)はデューティ比スイッチオンイベント処理動作を示すフローチャートである。デューティ比スイッチ44がオンされると、オンされたスイッチに応じてデータDRを増減するとともに、CPU占有率表示部40の点線40bをこれに応じて表示変更する(S75)。

【0043】図11はこの発明の他の適用例を示すフローチャートである。このフローチャートは、上記音源処理の他の例(第2実施態様)である。まず、この音源処理動作の打ち切り時刻を算出する(S81)。打ち切り時刻TL(タイムリミット)は、図7の場合と同様に、 $TL = ST + FL \times DR - US - AS$   
ST: 現再生フレームのスタート時刻  
FL: フレーム時間長(1フレーム全体の長さ)  
DR: デューティ比  
US: 打ち切り処理時間(打ち切り処理(フォースダンプ)に要する時間)  
AS: 後処理時間(他の処理に制御を渡すために要する時間)  
で計算される。

【0044】次にチャンネル制御を行う(S82)。チャンネル制御とは、優先度の高いチャンネルから先に演算するように32チャンネルの演算順序を設定する処理である。こののち、3つの独立した出力バッファOB1、OB2、OB3をクリアするとともに、演算順を示すポインタiに1をセットして(S83)、波形データ演算処理(S84～S92)を実行する。

【0045】まずi番目のチャンネルの音源レジスタにアドレスを設定するなどの波形データ演算準備処理を実行する(S84)。つぎに、サンプリング周波数制御レジスタCD5を判断する(S85)。CD5=2であれば、1フレーム(128サンプル)の波形データをそのまま形成するためS88に進んで、128サンプル分の波形データ演算処理を行い、OB2に加算する。CD5



=1であれば、1フレームを半分の精度（2倍周期のサンプリングクロック：64サンプル）で形成するため、S87に進んで、64サンプル分の波形データの演算処理を行い、OB1に加算する。また、CD5=0であれば、1フレームを1/4の精度（4倍周期のサンプリングクロック：32サンプル）で形成するため、S86に進んで、32サンプル分の波形データの演算処理を行い、OB0に加算する。

【0046】ここで、データCD5の値は、各チャンネル独立のデータCD5(i)として、各チャンネルの発音時にそれぞれの発音する楽音の音楽パートに応じて設定されるようにしてもよいし、逆に、ソフト音源全体として1つの値CD5を用いるようにしてもよい。例えば、1つの値にする場合、図4の制御画面中のサンプリング周波数設定表示部37にカーソルを置いてバリュースイッチ43を操作して設定する。設定された1つのデータCD5に応じて、S85における分岐が行われ、対応する処理が行われる。

【0047】一方、音楽パート毎の場合は、例えば、各音楽パートとしての各MIDIチャンネルの毎にデータCD5を設定する。すなわち、演奏に先立って演奏者は、1から16のMIDIチャンネルを指定し、対応する各パート毎のデータCD51~CD516の値を設定する。そして、ノートオンイベントが入力して図6のノートオンイベント処理の実行時に、ステップS23で実行されるicの音源レジスタの書き込みに際して、イベントの送られたパートtに応じた同チャンネルのデータCD5(i)を音源レジスタ内に設定すればよい。すなわち、パートtに設定されたデータCD5tを、割り当てられた発音チャンネルiの音源レジスタのデータCD5(i)として設定する。以上の処理により、各発音チャンネルi毎にデータCD5(i)が設定され、それに基づいてステップS85の分岐が各発音チャンネル毎に独立に行われる。

【0048】次に、データCD5により制御される楽音特性について述べる。図11の音源処理において、S85の分岐により、各発音チャンネルで1フレーム分の波形として生成する波形サンプル数が増加する。各発音チャンネルで1フレーム分として生成するサンプル数が多いほど、そのチャンネルの処理に時間がかかる。すなわち、生成サンプル数と処理時間には相関関係がある。本実施態様では、1秒間に生成するフレーム数は固定（48k/128=375フレーム）であり、フレーム毎の生成サンプル数の変化は1秒当たりの生成サンプル数（等価サンプリング周波数と呼ぶ）の変化に相当する。等価サンプリング周波数は生成する楽音の実質的なサンプリング周波数に相当し、サンプリングの定理によれば生成する楽音は等価サンプリング周波数の半分以下の周波数成分を有する。すなわち、等価サンプリング周波数が高いほど、生成する楽音のクオリティ（音質）は高

い。それぞれ、等価サンプリング周波数は、48kHz（CD5=2）、24kHz（CD5=1）、12kHz（CD5=0）である。

【0049】こののち、処理打ち切り時刻TLになったか否かを判断する（S89）。打ち切り時刻になった場合には、打ち切り処理を実行する（S91）。打ち切り処理は、波形データを演算できなかったチャンネルの楽音波形をフォースダンプするための波形を生成し、出力バッファに足し込む処理である。打ち切り時間になっていなければ発音中の全チャンネルについて処理を終了したかを判定する（S90）。まだ発音中の全チャンネルに関する処理が終了していなければ、iに1を加算して（S92）、S84に戻る。

【0050】全チャンネルに関する処理が終了し、または、打ち切り処理が行われた場合にはS93に進む。S93では、出力バッファOB0の内容を4倍オーバーサンプリングすることにより128サンプル化してOB2に足し込むとともに、出力バッファOB1の内容を2倍オーバーサンプリングすることにより128サンプル化してOB2に足し込む。こののち、バッファOB2に記憶されている波形データに対してリバーブ処理を施し（S94）、リバーブ効果の付与された出力バッファOB2の波形データを再生部に再生予約する（S95）。以上の動作により、CD5の設定によりサンプリング周波数を変更して楽音波形を生成することができ、CPU10の能力に合わせた楽音波形の生成が可能になる。

【0051】なお、第3の実施態様として、ソフト音源の最大発音数をデータCD6として32以下の数値に予め制限して設定しておき、図7のS41の打ち切りの判断を、現在の発音数がこの制限して設定された発音数に達したか否かによって行うように変更してもよい。すなわち、本実施態様では、図4の最大発音数設定部38により、最大発音数CD6を演奏者が設定し、ソフト音源は、その設定されたCD6を最大発音数として、それ以下の同時発音数で楽音生成を行う。この場合には、発音数でCPU10の占有率（デューティ比）を制限することができる。

【0052】さらに、第4の実施態様について説明する。上記第1実施態様および第2実施態様では、演奏者の設定したデューティ比DRにより、CPUの有する演算能力のうちのソフト音源の処理に使用してよい能力の割合の上限を制御している。一方、上記第3実施態様では、演奏者の設定した最大発音数CD6により、ソフト音源で生成する楽音の最大数が制限される。いずれの実施態様においても、ソフト音源で生成する楽音数が変化するのであるが、その変化は波形を生成する音源処理S8の内部だけであり、外部のプログラム、例えば、発音割当処理等ではその情報が生かされていない。

【0053】そこで、第4の実態様では、同時発音数の可変な音源の、ノートオンイベント処理を実行する際

に現在の同時発音数に応じて発音割当処理S21の割り当ての態様を変化させる。すなわち、ノートオンイベント処理のステップS21において、まず、データDRに応じたCPU演算能力を換算した同時発音数、ないし、データCD6を受け取り、受け取ったデータを最大発音数MPとして発音割り当てを行う。音源レジスタで現在発音中とされているチャンネルの数PNが、データ(MP-1)以下のときは、空いているチャンネルを割り当てチャンネルにするし、データMPより大きいときはその中からオーバーする分(PN-MP+1)個の消音チャンネルを決定して消音指示を当該チャンネルに書き込むとともに、1つの空きチャンネルを確保して、それを割り当てチャンネルとする。この処理では、現在の発音数に応じて発音割り当てを行っているの、発音を沢山割り当て過ぎてノイズが生じたり、少なく割り当てて折角の発音可能能力を活かし切れなかったりということを防止できる。音源の同時発音数は、音色の切替えによって各チャンネル当たりの処理量が増えたり、演奏中にデータDR、CD6を変化させたりすると、時間の経過でどんどん変化していくが、上述したように、ノートオンイベント等で発音割り当てを行うときに音源からの同時発音数を得るようにしておけば、それにも対応可能である。なお、この方式は、ソフト音源に限らず、状況に応じて最大発音数が増える全ての音源に適用可能である。

【0054】なお、上記実施態様では、ソフト音源全体について、デジタルフィルタDCFのオン/オフ設定を行うようにしているが、音色パート別、各発音チャンネル別に設定を行うようにしてもよい。すなわち、各音色パート別、各発音チャンネル別にデジタルフィルタ設定レジスタCD3(i)を設け、各音色パート、発音チャンネルの設定モードの中でこれをオン/オフ設定できるようにしてもよい。上記音源処理(図7)では発音チャンネル別に分岐処理S37を行っているため、これに対応する処理は容易である。また、その他のデータCD1、CD2等についても同様に、音色パート別、各発音チャンネル別に設定できるようにしてもよい。

【0055】リズム音を生成しようとする場合は、通常、ピッチ変換の必要がないため、CD2=0(補間なし)を設定するが、他の楽器に対してチューニングしたい場合には、CD2=1としてピッチ制御をするようにしてもよい。本実施態様では、MIDIイベント処理をデューティ比制御の計算にいられていないが、このMIDIイベント処理もソフト音源が側の動作としてデューティ比制御の計算にいられるようにしてもよい。

【0056】CPUパワーの表示や発音数は、高速で変動する場合があるので、見やすくするために表示のスムージングを行ってもよい。外部からのMIDI受信に限らず、コンピュータ自信の実行する自動演奏プログラムで再生されたMIDIイベントやゲームソフト等で発生

する発音指示等をソフト音源の演奏入力としてもよい。また、本発明のソフト音源は、上記実施態様で説明したように、Windows等のOSが搭載された汎用のコンピュータで実行してもよいが、これに限らず、鍵盤等の演奏操作子を備えた電子楽器や演奏操作子を備えない音源モジュール等の内部にある制御用CPUに実行させるようにしてもよい。この場合、従来電子回路で構成していた音源部を縮小または省略することができる。また、ハードの音源部とこのソフト音源を併用してもよい。

【0057】本実施態様のエフェクタ制御では、データCD4の値に応じて、エフェクト無し、ローパスフィルタ、リバース処理の3つの中から1つを選択するようになっている。すなわち、ローパスフィルタ、リバース処理という、効果の内容が異なるエフェクトを選択的に付与できるようになっている。または、処理に必要な演算量が異なるエフェクトの選択と見ることもできる。このエフェクトの選択に関しては、さらに、上記実施態様以外の態様が考えられる。例えば、同一種類のエフェクトであるが、エフェクトの処理のグレード、および、処理に必要な演算量の異なる複数のエフェクトプログラムから1つを選択するようにしてもよい。すなわち、エフェクトのグレードを上げれば、その分、他に回せる演算量が減り、同時に生成できるチャンネル数の減少を引き起こす。演奏者はCD4を設定することにより、演奏曲等に応じて同時発音数とエフェクトグレードのバランスをとることができる。

【0058】また、図2に示したソフト音源の時間的な処理の流れは、1つの例であり、フレームのサイズや、演奏入力、生成演算、読出再生の時間的な相互関係はこの図に限定されない。例えば、演奏入力、生成演算、読出再生のそれぞれの時間フレームの時間切り替わり位置が異なるようにしてもよい。さらに、1フレームの時間間隔が固定長になっているが、これを毎フレームに異なる長さになるようにしてもよい。ただし、1回の音源処理で、時間方向に複数の波形サンプルを生成するという点は、本願発明の実施態様の特徴である。

【0059】

【発明の効果】以上の通り、この発明によれば、プログラムに従って動作するCPUにより、演奏情報の受信と選択情報の受信が行なわれ、受信した演奏情報に基づく楽音生成処理が行なわれるとともに、選択情報に応じて該楽音生成における所定の特性制御処理を追加または省略することにより、当該CPUで同時に生成可能な最大発音数を変化させることができ、かつ、同一の演奏情報に対しても異なる演算処理が実行され得るようになるので、選択情報に応じて必要な特性制御処理を追加した必要な特性制御処理を削除することにより、その都度の楽音生成に適した楽音波形生成制御が可能になり、また、不要な処理を削減することによって演算量が軽減されるた

め、発音数を増加したり、楽音生成以外の処理を行うこともできる。従って、生成しようとする楽音の目的や演奏形態に適合させて、楽音制御の内容を変更したり、特定の楽音制御をやめることで処理量を軽減し、その分発音数を増加させたり、他の目的にCPUのパワーをまわしたりすることができる。これによって、例えば、マイクロコンピュータを用いたソフト音源を実施する場合、コンピュータ装置のCPUパワーを有効に活用することができる。また、他のアプリケーションプログラムと並行してソフトウェア音源プログラムを実行する場合、同時に生成可能な最大発音数を変化させたり、同一の演奏情報に対しても異なる演算処理が実行され得るようにすることで、ソフト音源側の演算量を調整することができるので、並行して動作する他のアプリケーションプログラムに対して悪影響を与えないようにすることもできる。従って、ソフト音源と並行して他のアプリケーションを安定して動作させる場合にも適している。

【0060】また、この発明によれば、プログラムに従って動作するCPUが、演奏情報を受信するとともに当該CPUの演算量の上限を示す制限情報を受信し、前記演奏情報に基づく演算処理によって楽音を生成するとともに、前記CPUで同時に生成する楽音数を、前記制限情報が示す演算量の上限に応じて定まる最大同時発音数以下になるように各楽音の優先度に従って制限することを特徴としているので、演算量を制限しつつも、その限られた演算量のなかで生成できた楽音については音切れ等起こさず安定した発音が行える。また、発音数の制限を行なうに際し、各楽音の優先度に従って制限するので、優先度の高い楽音が消音されないこととなり、非常に好ましい。従って、他のアプリケーションプログラムと並行してソフトウェア音源プログラムを実行する場合、ソフト音源がCPUパワーを占有する時間を制限情報で制限することにより、円滑なマルチタスク状態を確保するのに適している。

【0061】さらに、この発明によれば、プログラムに従って動作するCPUが、演奏情報を受信するとともに任意に設定されたサンプリング周波数を示す演算精度情報を受信し、前記演奏情報に基づき、前記演算精度情報が示すサンプリング周波数の演算処理によって楽音を生

成することを特徴とするものであるから、サンプリング周波数を任意に設定することによって演算量を任意に変設定することができ、これによって、臨機応変に、効率的なCPU資源の利用を図ることができる。また、波形補間の種類を示す演算精度情報に応じて楽音生成処理における波形補間の精度を可変制御することによっても、演算量の増減制御を行なうことができるので、効率的なCPU資源の利用を図ることができる。

#### 【図面の簡単な説明】

【図1】 この発明が適用されるマイクロコンピュータ装置のブロック図。

【図2】 同マイクロコンピュータ装置で動作するソフト音源の楽音生成処理の時間的流れを説明する図。

【図3】 同ソフト音源実行時にRAMに設定されるメモリエリアを説明する図。

【図4】 同ソフト音源実行時に表示される画面を示す図。

【図5】 同ソフト音源の動作を示すフローチャート。

【図6】 同ソフト音源の動作を示すフローチャート。

【図7】 同ソフト音源の動作を示すフローチャート。

【図8】 同ソフト音源の動作を示すフローチャート。

【図9】 同ソフト音源の動作を示すフローチャート。

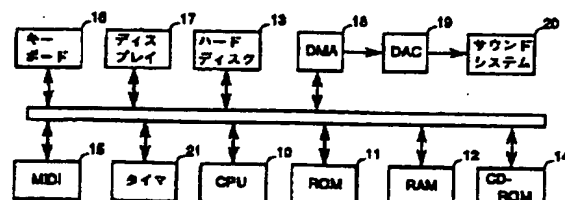
【図10】 同ソフト音源の動作を示すフローチャート。

【図11】 同ソフト音源の他の動作例を示すフローチャート。

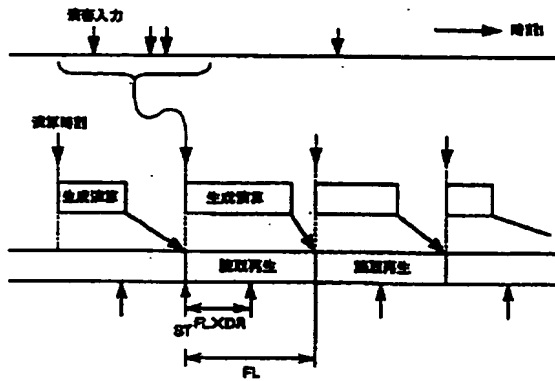
#### 【符号の説明】

- 10 CPU
- 12 RAM
- 14 CD-ROM
- 15 MIDIインタフェース
- 18 DMA
- 19 DAC
- 20 サウンドシステム
- 30 制御パネルウィンドウ
- 33～38 設定表示部
- 39 現在発音数表示部
- 40 デューティ比表示部
- 44 デューティ比スイッチ

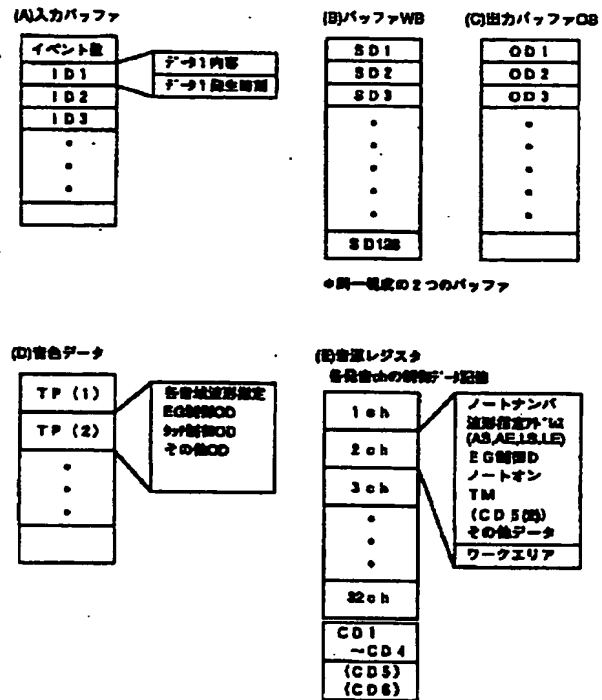
【図1】



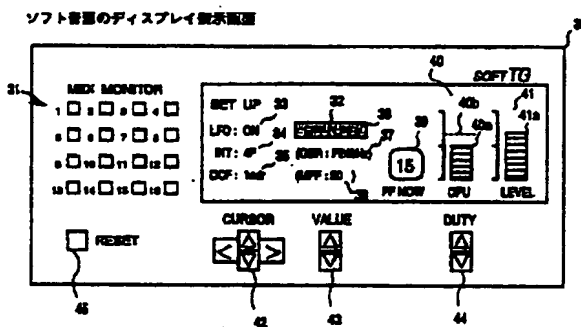
【図 2】



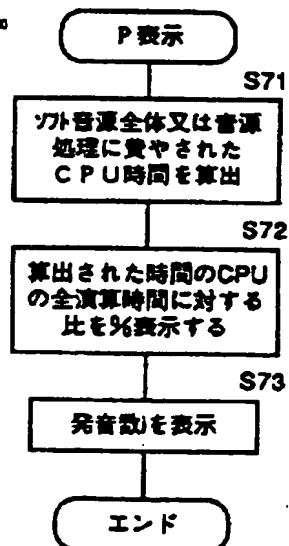
【図 3】



【図 4】

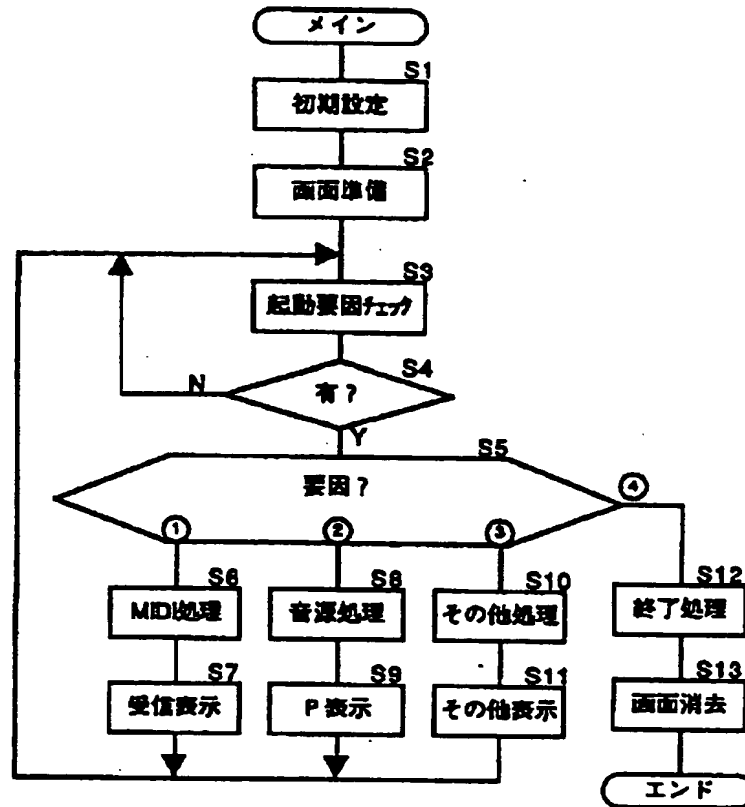


【図 9】

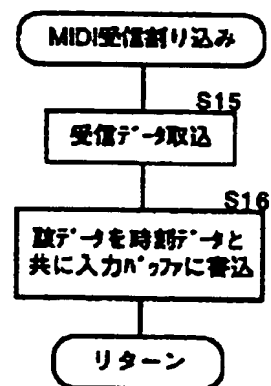


【図5】

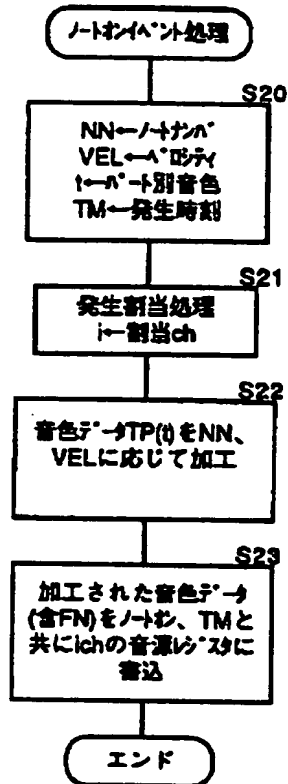
(A)



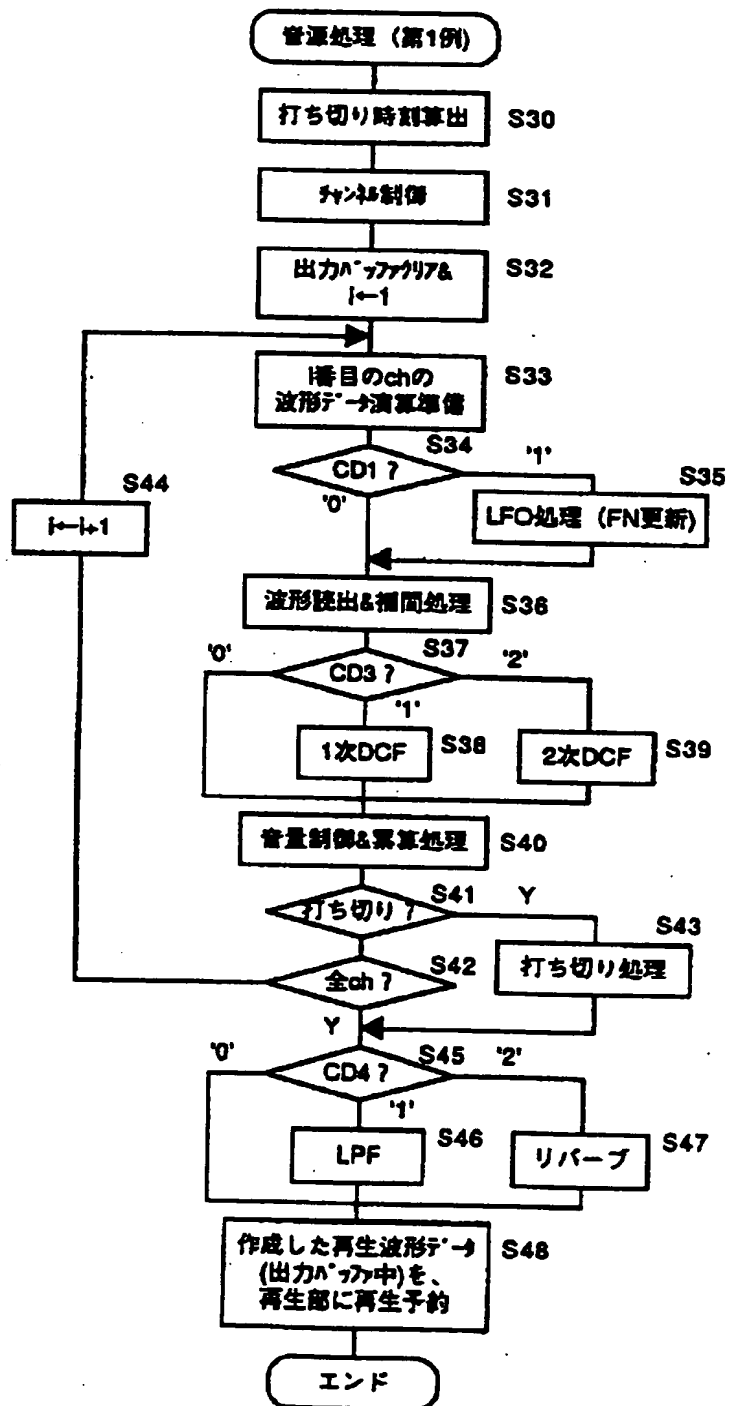
(B) 最優先の割り込み処理



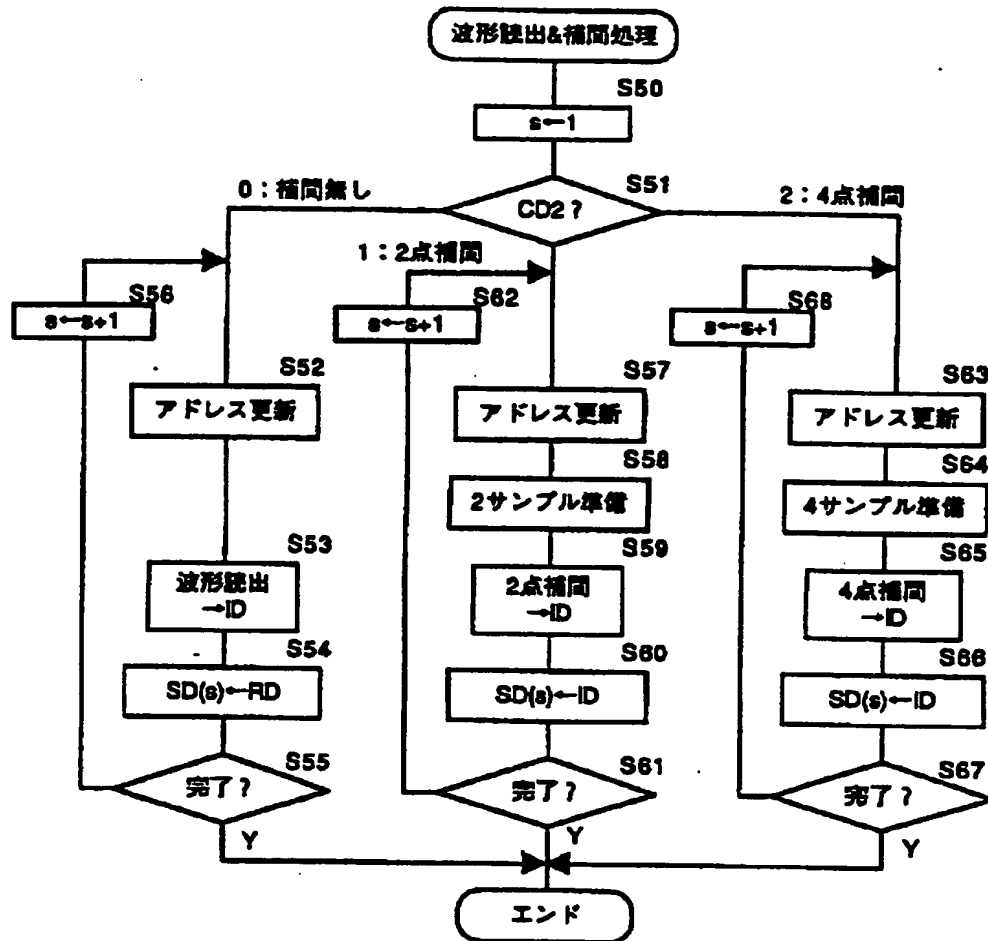
【図6】



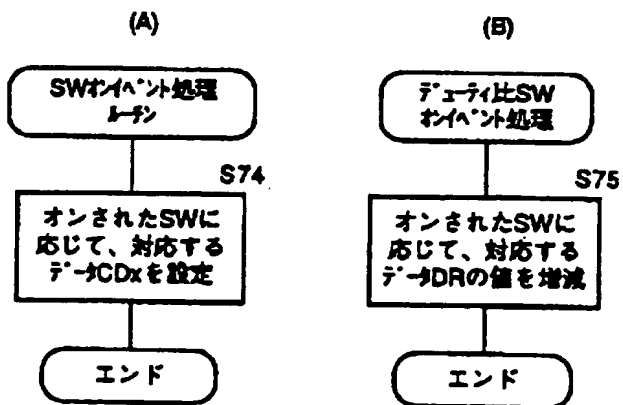
【図7】



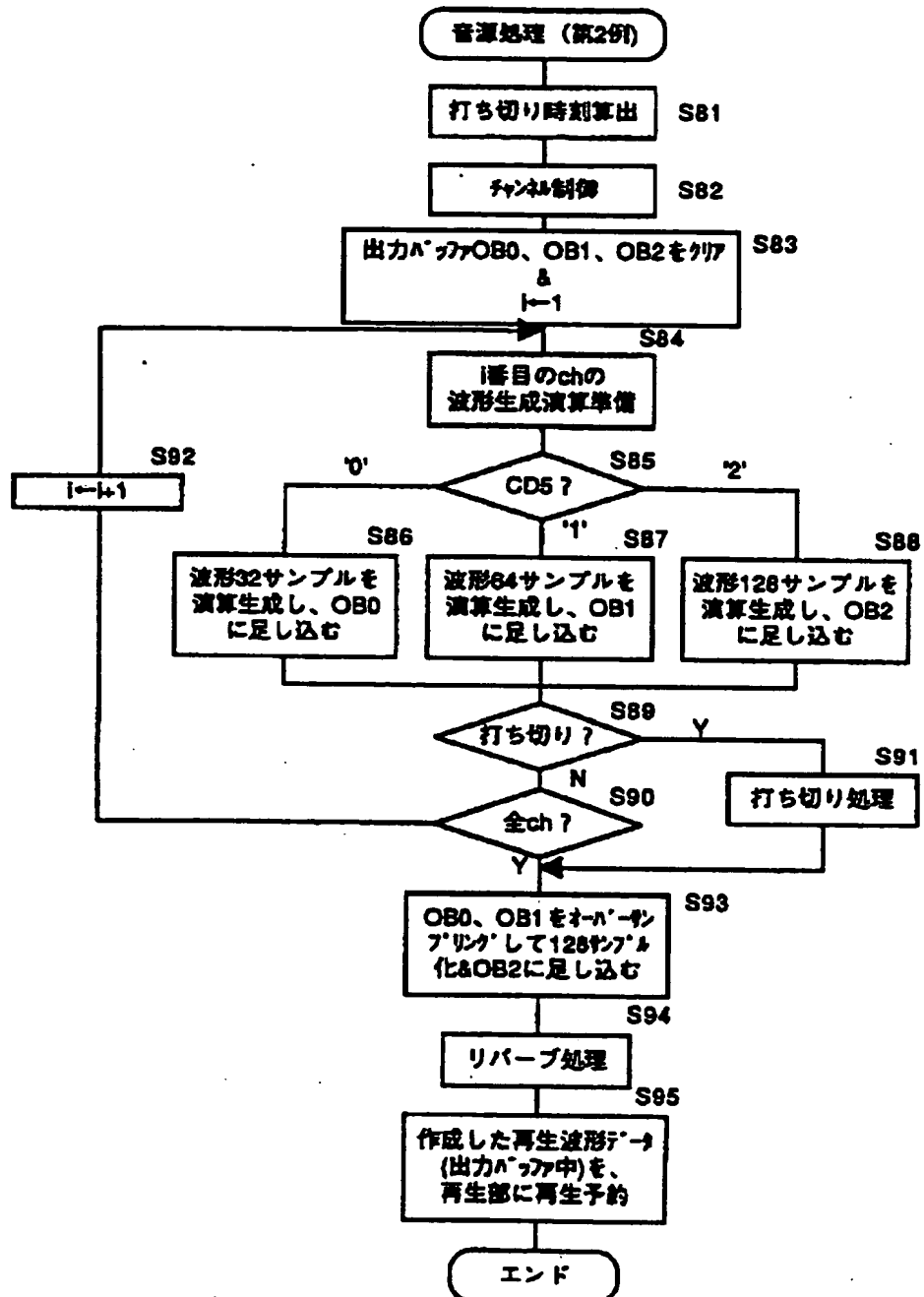
【図 8】



【図 10】



【図 1 1】





**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**